A Nonparametric Riemannian Framework for Processing High Angular Resolution Diffusion Images (HARDI)

Alvina Goh[1] Christophe Lenglet[2] Paul M. Thompson[3] René Vidal[1]

[1]Johns Hopkins University [2]University of Minnesota [3]University of California, Los Angeles

{agoh,rvidal}@jhu.edu clenglet@umn.edu thompson@loni.ucla.edu

Abstract

High angular resolution diffusion imaging has become an important magnetic resonance technique for in vivo imaging. Most current research in this field focuses on developing methods for computing the orientation distribution function (ODF), which is the probability distribution function of water molecule diffusion along any angle on the sphere. In this paper, we present a Riemannian framework to carry out computations on an ODF field. The proposed framework does not require that the ODFs be represented by any fixed parameterization, such as a mixture of von Mises-Fisher distributions or a spherical harmonic expansion. Instead, we use a non-parametric representation of the ODF, and exploit the fact that under the square-root re-parameterization, the space of ODFs forms a Riemannian manifold, namely the unit Hilbert sphere. Specifically, we use Riemannian operations to perform various geometric data processing algorithms, such as interpolation, convolution and linear and nonlinear filtering. We illustrate these concepts with numerical experiments on synthetic and real datasets.

1. Introduction

Diffusion magnetic resonance imaging (MRI) is an imaging technique that produces in vivo images of biological tissues by exploiting the constrained diffusion properties of water molecules. Water diffusion is hindered less along fibrous structures as the molecules encounter fewer tissue barriers when they move in that direction. The signal intensity at each position depends on the local microstructure in which the water molecules diffuse as well as the strength and direction of the applied magnetic field gradients. Multiple MR scans with varying gradient directions and strengths are used to capture the complete diffusion profile of a tissue. As the directions of maximum diffusion indicate the structural anisotropy of the medium, diffusion MRI can be used to infer the organization and orientation of tissue components. Experiments have shown that water diffusion is indeed anisotropic in organized tissues such as the spinal cord, muscles, heart or brain white matter. This has generated much enthusiasm and high expectations, because diffusion MRI is presently the only available approach to non-invasively study the three-dimensional architecture of human tissues, and quantify their physical and geometrical properties.

Several techniques can be used to reconstruct orientation distribution functions from diffusion MRI. One classical technique is known as diffusion tensor imaging (DTI) [4]. In DTI, the diffusivity profile is characterized by a single oriented 3D Gaussian probability distribution function. Water diffusion is then represented mathematically with a symmetric positive semi-definite (SPSD) tensor field $\mathbf{D} : \mathbb{R}^3 \to \text{SPSD}(3) \subset \mathbb{R}^{3\times 3}$ that measures the extent of diffusion in a direction $\boldsymbol{v} \in \mathbb{R}^3$ as $\boldsymbol{v}^\top \mathbf{D} \boldsymbol{v}$. Even though DTI assumes a relatively simple diffusion model, it has been a successful imaging technique in regions of the brain and spinal cord with significant white-matter coherence. For instance, DTI has enabled the mapping of anatomical connections in the central nervous system [5, 16]. However, DTI models the diffusion with a single tensor, so it can only reveal a single fiber orientation within each voxel. When fibers cross or bifurcate within a voxel, which is common in complex brain structures, DTI will not be able to provide accurate insights into the diffusivity profile or fiber geometry.

Recent advances in diffusion MRI address this well-known limitation of DTI. In particular, an imaging method known as high angular resolution diffusion imaging (HARDI) has been proposed [30]. HARDI measures diffusion along n uniformly distributed directions on the sphere and is able to characterize complex fiber geometries. Several reconstruction techniques can be used to characterize diffusion based on the obtained HARDI signal. A common approach is to construct the orientation distribution function (ODF) from these measurements. The ODF is a *probability distribution function* that quantifies the probability of water diffusion along different directions on the sphere. A significant amount of the current research on HARDI focuses on computing the ODF from HARDI signals. One of the earliest methods uses the Funk-Radon transform to estimate ODFs [31]. In addition, spherical harmonic expansions have also been used to approximate ODFs [9, 12, 24]. Such methods

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

are typically very fast as the ODF is computed analytically. Finally, higher-order tensors, which can leverage the work done in DTI, have also been used to model diffusivity [3, 14].

While much of the current research efforts focus on computing the ODFs efficiently and accurately, much less work has been done on the processing of these high-dimensional datasets. That is, given a set of ODFs, how would one perform statistical and principal components analysis, or basic operations such as interpolation, filtering and convolution? Unlike in DTI where the geometry of the SPSD space is well studied and several frameworks [2, 18, 21, 25] have been proposed, such frameworks are less common for HARDI data. In order to perform any operations, one will first need a metric to compare different ODFs. As in DTI, this is an open issues and several choices, such as Kullback-Leibler divergence (KLD) [10], Jensen-Shannon divergence (JSD) [8] and Riemannian distance between a mixture of von Mises-Fisher distributions [22], have been proposed. KLD is difficult to compute for the spherical data with high angular resolution and it also has to be symmetrized. Both the symmetrized version of KLD (SKLD) and SJD do not obey the triangular inequality and therefore are not metrics. The work of [22] presents a Riemannian metric and a possible framework for HARDI processing. In their work, the precomputed ODF is represented by a mixture of von Mises-Fisher distributions, where the mixture parameters are computed using an expectation-maximization algorithm. Once that is done, the authors show that it is possible to compute distances and various Riemannian operations between different mixtures of von Mises-Fisher distributions using closed form formulae. However, this framework suffers from two major issues. First, the number of components in the mixture model is arbitrarily chosen. To find the optimal number of components that one should use, the difficult question of model selection must be addressed. The second issue is that a nonlinear least-squares technique is used to compute the unknown parameters. This technique may not be stable as the number of parameters increases, when the complexity of the tissue increases. Therefore, a framework is needed to allow the processing of ODFs without having to resort to such models, while respecting their intrinsic mathematical properties.

Paper Contributions: In this paper, we present a Riemannian framework for the processing of ODFs. We operate directly on the precomputed ODF and do not require that the ODF be represented by any parametric model such as a mixture of von Mises-Fisher distributions. We exploit the fact that ODFs are probability density functions defined on the 2-sphere S^2, and that under the square-root re-parametrization, the space of ODFs forms a Riemannian manifold, namely the unit Hilbert sphere. Therefore, various Riemannian operations such as the exponential map, logarithmic map and geodesics are not only available in closed form, but also easily and efficiently computed. We use these operations to introduce various geometric data processing algorithms such as interpolation, convolution, and linear and nonlinear filtering. We present experiments on synthetic and real data.

2. The Riemannian Manifold of Orientation Distribution Functions

In this section, we show that the orientation distribution functions lie on a Riemannian manifold. We first present a brief summary of the theory of Riemannian manifolds. We then illustrate the Riemannian structure of the space of ODFs, under the square-root representation, which gives closed-form expressions for various Riemannian operations.

2.1. Review of Riemannian Manifolds

A differentiable manifold $\mathcal{M}$ of dimension d is a topological space that is homeomorphic to the Euclidean space $\mathbb{R}^d$. The tangent space $T_{\boldsymbol{x}}\mathcal{M}$ at $\boldsymbol{x}$ is the vector space that contains the tangent vectors to all 1-D curves on $\mathcal{M}$ passing through $\boldsymbol{x}$. A *Riemannian metric* on a manifold $\mathcal{M}$ is a bilinear form that associates to each point $\boldsymbol{x} \in \mathcal{M}$, a differentiable varying inner product $\langle \cdot, \cdot \rangle_{\boldsymbol{x}}$ on the tangent space $T_{\boldsymbol{x}}\mathcal{M}$ at $\boldsymbol{x}$. The norm of a vector $\boldsymbol{v} \in T_{\boldsymbol{x}}\mathcal{M}$ is denoted by $\|\boldsymbol{v}\|_{\boldsymbol{x}}^2 = \langle \boldsymbol{v}, \boldsymbol{v} \rangle_{\boldsymbol{x}}$. The Riemannian distance $\mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ between two points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ lying in the manifold is defined as the minimum length over all possible smooth curves on the manifold between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The geodesic curve from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$, γ, is the smooth curve with minimum length.

Given a tangent vector $\boldsymbol{v} \in T_{\boldsymbol{x}}\mathcal{M}$, there exists a unique geodesic $\gamma_{\boldsymbol{v}}(t)$ starting at $\boldsymbol{x}$ with initial velocity $\boldsymbol{v}$, and this geodesic has constant speed equal to $\|\boldsymbol{v}\|_{\boldsymbol{x}}$. The *exponential map*, $\exp_{\boldsymbol{x}} : T_{\boldsymbol{x}}\mathcal{M} \rightarrow \mathcal{M}$ maps a tangent vector $\boldsymbol{v}$ to the point on the manifold that is reached at time 1 by the geodesic $\gamma_{\boldsymbol{v}}(t)$. The inverse of $\exp_{\boldsymbol{x}}$ is the *logarithm map* and denoted by $\log_{\boldsymbol{x}} : \mathcal{M} \rightarrow T_{\boldsymbol{x}}\mathcal{M}$. For two points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ on the manifold $\mathcal{M}$, the tangent vector to the geodesic curve from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ is defined as $\boldsymbol{v} = \overrightarrow{\boldsymbol{x}_i \boldsymbol{x}_j} = \log_{\boldsymbol{x}_i}(\boldsymbol{x}_j)$, and the exponential map takes $\boldsymbol{v}$ to the point $\boldsymbol{x}_j = \exp_{\boldsymbol{x}_i}(\log_{\boldsymbol{x}_i}(\boldsymbol{x}_j))$. In addition, $\gamma_{\boldsymbol{v}}(0) = \boldsymbol{x}_i$ and $\gamma_{\boldsymbol{v}}(1) = \boldsymbol{x}_j$. The Riemannian distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is defined as $\mathrm{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\log_{\boldsymbol{x}_i}(\boldsymbol{x}_j)\|_{\boldsymbol{x}_i}$. Linear geodesic interpolation makes use of the exponential and logarithm maps and is defined as $\widehat{\boldsymbol{x}} = \exp_{\boldsymbol{x}_i}(w\overrightarrow{\boldsymbol{x}_i \boldsymbol{x}_j})$, $w \in [0, 1]$. $\widehat{\boldsymbol{x}}$ is the linear interpolation at $t = w$ of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Finally, we recall that the Riemannian metric, exponential and logarithm maps depend on the point $\boldsymbol{x}$ under consideration, hence the subscripts reflecting this dependency.

2.2. Parameterization of ODFs

We will now show how to impose a Riemannian structure on the space of ODFs or equivalently, the space of probability density functions (PDF). In particular, we will adopt a "spherical" version of the Fisher-Rao metric that allows for closed-form expressions of the various Riemannian operations. The class of constrained non-negative functions under

study here is the set of probability density functions on the 2-sphere S^2. Specifically, we consider the set of PDFs

$$\boldsymbol{\mathcal{P}} = \{\boldsymbol{p} : S^2 \rightarrow \mathbb{R}^+ | \forall \boldsymbol{s} \in S^2, \boldsymbol{p}(\boldsymbol{s}) \geq 0; \sum_{\boldsymbol{s} \in S^2} \boldsymbol{p}(\boldsymbol{s}) = 1\}.$$

The question of how to equip the space of PDFs with a differential manifold structure, a Riemannian metric and a family of affine connections has a long history. Nevertheless, it remains an active and important research area. Treating statistical quantities as geometric structures has the advantage of preserving the invariance of such structures under change of coordinates. Studying probability and information via differential geometry is known as *information geometry*. The reader is referred to [1] for a complete description. Rao [27] first introduced the Riemannian structure formed by the statistical manifold whose elements are probability density functions. In addition, he also showed [27] that the *Fisher-Rao* metric determines a Riemannian metric. The Fisher-Rao metric is defined as

$$\langle \boldsymbol{q}_j, \boldsymbol{q}_k \rangle_{\boldsymbol{p}_i} = \sum_{\boldsymbol{s} \in S^2} \boldsymbol{q}_j(\boldsymbol{s}) \boldsymbol{q}_k(\boldsymbol{s}) \frac{1}{\boldsymbol{p}_i(\boldsymbol{s})}, \tag{1}$$

where $\boldsymbol{q}_j, \boldsymbol{q}_k \in T_{\boldsymbol{p}_i}\boldsymbol{\mathcal{P}}$ are tangent vectors and $T_{\boldsymbol{p}_i}\boldsymbol{\mathcal{P}}$ is the set containing the functions tangent to $\boldsymbol{\mathcal{P}}$ at the point $\boldsymbol{p}_i$. This metric was later shown to be the *unique intrinsic metric* on the statistical manifold in [7], therefore invariant to re-parameterizations (essentially coordinate transforms) of the functions. However, this representation turns out to be extremely difficult to work with as the computation of the geodesic between two elements is not easy [29].

Even though the space $\boldsymbol{\mathcal{P}}$ turns out to be difficult to work with, we know that it is not the only possible representation for PDFs. There are many different re-parameterizations of PDFs that are equivalent. These include the cumulative distribution function, log density function and square-root density function. Each of these parameterizations will lead to a different resulting manifold. Depending on the choice of representation, the resulting Riemannian structure can have varying degrees of complexity and numerical techniques may be required to compute geodesics on the manifold. For example, the authors of [23] chose the log density representation and used a shooting technique to find geodesics on this space. However, this space has a complicated Riemannian structure and the numerical method used in [23] sometimes leads to large errors. Therefore, a natural question to ask is: Is it possible to use a re-parameterization such that the resulting manifold is simple and the Riemannian operations are easy to compute, preferably in closed-form? Once an efficient representation is found, the corresponding Fisher-Rao metric, which depends on the tangent vector, can be used as the Riemannian metric.

In this paper, the choice of parameterization is the square-root representation. This representation has been used in [15, 20, 29]. We will consider the space of ODFs, which are discrete multi-variate probability density functions. The main reason behind choosing the square-root representation is that the resulting manifold is a unit sphere in a Hilbert space with the Fisher-Rao metric being the usual $\mathbb{L}^2$ metric. Therefore, the various Riemannian operations such as geodesics, exponential maps, logarithmic maps are available in closed form.

The square-root density function is one of the most efficient representations found as of today and is defined as

$$\boldsymbol{\psi}(\boldsymbol{s}) = \sqrt{\boldsymbol{p}(\boldsymbol{s})},$$

where $\boldsymbol{\psi}(\boldsymbol{s})$ is assumed to be non-negative to ensure uniqueness. The space of such functions is defined as:

$$\boldsymbol{\Psi} = \{\boldsymbol{\psi} : S^2 \rightarrow \mathbb{R}^+ | \forall \boldsymbol{s} \in S^2, \boldsymbol{\psi}(s) \geq 0; \sum_{\boldsymbol{s} \in S^2} \boldsymbol{\psi}^2(\boldsymbol{s}) = 1\}. \tag{2}$$

From Eq. (2), it is easy to see that the functions $\boldsymbol{\psi}$ lie on the positive orthant of a unit Hilbert sphere[1]. In addition, $\boldsymbol{\Psi}$ forms a *convex subset* of the unit Hilbert sphere. The advantage of choosing the square-root density becomes immediately obvious, as many of the Riemannian expressions for the unit Hilbert sphere are well-known and closed-form expressions. Now, making use of the representation in Eq. (2) and the fact that Riemannian metrics are determined up to a constant scaling factor, we can rewrite Eq. (1) and obtain the Fisher-Rao metric as

$$\langle \boldsymbol{\phi}_j, \boldsymbol{\phi}_k \rangle_{\boldsymbol{\psi}_i} = \sum_{\boldsymbol{s} \in S^2} \boldsymbol{\phi}_j(\boldsymbol{s}) \boldsymbol{\phi}_k(\boldsymbol{s}),$$

where $\boldsymbol{\phi}_j, \boldsymbol{\phi}_k \in T_{\boldsymbol{\psi}_i}\boldsymbol{\Psi}$ are tangent vectors.

2.3. Geodesic Marching on the Space $\boldsymbol{\Psi}$

In this section, we will show how to compute the various Riemannian operations in $\boldsymbol{\Psi}$ in order to evolve (or "march") along its geodesic curves. For any two functions $\boldsymbol{\psi}_i, \boldsymbol{\psi}_j \in \boldsymbol{\Psi}$, the geodesic distance between these two points on a unit Hilbert sphere is simply the angle between them,

$$\begin{aligned} \text{dist}(\boldsymbol{\psi}_i, \boldsymbol{\psi}_j) &= \| \log_{\boldsymbol{\psi}_i}(\boldsymbol{\psi}_j) \|_{\boldsymbol{\psi}_i} \\ &= \cos^{-1} \langle \boldsymbol{\psi}_i, \boldsymbol{\psi}_j \rangle = \cos^{-1} \left(\sum_{\boldsymbol{s} \in S^2} \boldsymbol{\psi}_i(\boldsymbol{s}) \boldsymbol{\psi}_j(\boldsymbol{s}) \right), \end{aligned} \tag{3}$$

where $\langle \cdot, \cdot \rangle$ is the normal dot product between points in the sphere under the $\mathbb{L}^2$ metric. From the differential geometry of the sphere, the exponential map is defined as

$$\exp_{\boldsymbol{\psi}_i}(\boldsymbol{\phi}) = \cos(\|\boldsymbol{\phi}\|_{\boldsymbol{\psi}_i}) \boldsymbol{\psi}_i + \sin(\|\boldsymbol{\phi}\|_{\boldsymbol{\psi}_i}) \frac{\boldsymbol{\phi}}{\|\boldsymbol{\phi}\|_{\boldsymbol{\psi}_i}}, \tag{4}$$

where $\boldsymbol{\phi} \in T_{\boldsymbol{\psi}_i}\boldsymbol{\Psi}$ is a tangent vector at $\boldsymbol{\psi}_i$ and

$$\|\boldsymbol{\phi}\|_{\boldsymbol{\psi}_i} = \sqrt{\langle \boldsymbol{\phi}, \boldsymbol{\phi} \rangle_{\boldsymbol{\psi}_i}}.$$

In order to ensure that the exponential map is bijective and stays on the positive orthant, we restrict $\|\boldsymbol{\phi}\|_{\boldsymbol{\psi}_i} \in [0, \frac{\pi}{2}]$.

[1] Note that the usage of the word "sphere" might be confusing. ODFs are defined on the 2-sphere S^2, whereas in Eq. (2), each $\boldsymbol{\psi}$ is a point on the unit Hilbert sphere.

The logarithm map from $\boldsymbol{\psi}_i$ to $\boldsymbol{\psi}_j$ is then given by

$$\overrightarrow{\boldsymbol{\psi}_i\boldsymbol{\psi}_j} = \log_{\boldsymbol{\psi}_i}(\boldsymbol{\psi}_j) = \frac{\boldsymbol{\psi}_j - \langle \boldsymbol{\psi}_i, \boldsymbol{\psi}_j \rangle \boldsymbol{\psi}_i}{\sqrt{1 - \langle \boldsymbol{\psi}_i, \boldsymbol{\psi}_j \rangle^2}} \cos^{-1}\langle \boldsymbol{\psi}_i, \boldsymbol{\psi}_j \rangle. \tag{5}$$

2.4. Statistics on the Space $\boldsymbol{\Psi}$

We will now briefly illustrate how to calculate the mean and principal components of points lying on $\boldsymbol{\Psi}$. As defined by Fréchet in [13], the intrinsic mean $\overline{\boldsymbol{\psi}}$ is the solution to the following minimization problem

$$\overline{\boldsymbol{\psi}} = \arg\min_{\boldsymbol{\psi} \in \boldsymbol{\Psi}} \frac{1}{n} \sum_{i=1}^{n} \text{dist}(\boldsymbol{\psi}, \boldsymbol{\psi}_i)^2. \tag{6}$$

Note that, unlike in the Euclidean case, in general there is no closed form for the intrinsic mean on Riemannian manifolds. Moreover, there is no guarantee that it exists or is unique. However, one can prove the existence and uniqueness of the mean [17] when the manifold has non-positive sectional curvature or by assuming that the data lies in a small enough neighborhood. Even though the mean might not be well-defined for a generic Riemannian manifold (consider two antipodal points on a sphere for instance), it is unique for a convex subset of a Riemannian manifold [6]. This is true despite that the sectional curvature of the unit Hilbert sphere having a positive value of 1. Therefore, the mean *is unique and exists for* $\boldsymbol{\Psi}$, the space of ODFs. Similarly to Eq. (6), the weighted mean is defined as

$$\overline{\boldsymbol{\psi}} = \arg\min_{\boldsymbol{\psi} \in \boldsymbol{\Psi}} \sum_{i=1}^{n} w_i \text{dist}(\boldsymbol{\psi}, \boldsymbol{\psi}_i)^2.$$

where $w_i \geq 0$ and $\sum_{i=1}^{n} w_i = 1$, in order to ensure that the solution exists and is unique. Notice that when $w_i = \frac{1}{n}$, we have the intrinsic mean. In addition, the (weighted) mean is characterized by being the unique solution to

$$\sum_{i=1}^{n} w_i \log_{\overline{\boldsymbol{\psi}}}(\boldsymbol{\psi}_i) = \mathbf{0}. \tag{7}$$

Since the exponential and logarithm maps are known (Eq. (4) and (5)), it is possible to compute $\overline{\boldsymbol{\psi}}$ using geodesic marching, as detailed in Algorithm 1.

Algorithm 1 (Weighted Mean)

Given data points $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_n \in \boldsymbol{\Psi}$, a predefined threshold ϵ, maximum number of iterations T,

1. Initialize $t = 1$, $\overline{\boldsymbol{\psi}}_1 = \boldsymbol{x}_i$ for a random i.
2. While $t \leq T$ or $\|\boldsymbol{\phi}\|_{\overline{\boldsymbol{\psi}}} \geq \epsilon$,
   (a) Compute tangent vector $\boldsymbol{\phi} = \sum_{i=1}^{n} w_i \log_{\overline{\boldsymbol{\psi}}_t}(\boldsymbol{\psi}_i)$,
   (b) Set $\overline{\boldsymbol{\psi}}_{t+1} = \exp_{\overline{\boldsymbol{\psi}}_t}(\boldsymbol{\phi})$.

Given $\overline{\boldsymbol{\psi}}$, the calculation of principal components on a Riemannian manifold is not as straightforward as in the Euclidean case. In [11], it is shown that finding principal components boils down to doing principal components analysis (PCA) on the tangent vectors $\log_{\overline{\boldsymbol{\psi}}}(\boldsymbol{\psi}_i) \in T_{\overline{\boldsymbol{\psi}}}\boldsymbol{\Psi}$ about the mean $\overline{\boldsymbol{\psi}}$. This algorithm, known as Principal Geodesic Analysis (PGA), is summarized in Algorithm 2.

Algorithm 2 (Principal Geodesic Analysis)

Given data points $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_n \in \boldsymbol{\Psi}$,

1. Compute intrinsic mean $\overline{\boldsymbol{\psi}}$ as in Algorithm 1.
2. Calculate the tangent vectors $\boldsymbol{\phi}_i = \log_{\overline{\boldsymbol{\psi}}}(\boldsymbol{\psi}_i)$about $\overline{\boldsymbol{\psi}}$.
3. Construct the sample covariance matrix $\text{cov}(\boldsymbol{\psi}) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\phi}_i \boldsymbol{\phi}_i^\top$.
4. Perform eigenanalysis of the matrix $\text{cov}(\boldsymbol{\psi})$, with the eigenvectors $\{\mathbf{u}_i\}_{i=1}^{d}$ giving the principal directions. $\{\mathbf{u}_i\}_{i=1}^{d}$ forms an orthonormal basis for $T_{\overline{\boldsymbol{\psi}}}\boldsymbol{\Psi}$.

3. Processing of ODFs Fields

In this section, we show how to process a field of ODFs using the tools of §2. In particular, we consider operations such as interpolation, convolution, and filtering of ODFs.

3.1. Spatial Interpolation of ODFs

Interpolation is one of the most important operations in data processing. In general, almost every geometric transformation requires interpolation to be performed on an image or a volume, e.g., translating, rotating, scaling, warping. The fundamental question that interpolation raises is that of the estimation of unknown data, by using known data. Standard interpolation methods are often based on attempts to generate continuous data from a set of discrete data samples through an interpolation function. To the best of our knowledge, spatial interpolation of ODFs is a new issue [8]. While trilinear interpolation has been previously used in [8], we present a complete overview of how one does spatial interpolation on both regular and non-regular grids here.

Linear interpolation

We will first start with the simplest case, namely linear interpolation. Let $x_1, x_2 \in \mathbb{R}$ and consider the interval $[x_1, x_2]$. Assume we know the ODFs at x_1 and x_2, $\boldsymbol{\psi}(x_1)$ and $\boldsymbol{\psi}(x_2)$ respectively, and we want to estimate unknown $\boldsymbol{\psi}$ at point $x \in [x_1, x_2]$. It is well-known that $\boldsymbol{\psi}(x) = \boldsymbol{\psi}(x_1) + w\overrightarrow{\boldsymbol{\psi}(x_1)\boldsymbol{\psi}(x_2)}$, $w = \frac{x - x_1}{x_2 - x_1}$ is the linear interpolation equation in Euclidean space. Therefore, for the Riemannian manifold $\boldsymbol{\Psi}$, we again have the closed-form expression $\boldsymbol{\psi}(x) = \exp_{\boldsymbol{\psi}(x_1)}(w\overrightarrow{\boldsymbol{\psi}(x_1)\boldsymbol{\psi}(x_2)})$.

Interpolation on a regular grid

We will now move on to the slightly more sophisticated processing tools commonly used in image processing: multi-dimensional interpolation such as bilinear and trilinear interpolation. These are extensions of linear interpolation to interpolating functions on a n-dimensional regular grid; $n = 2$ for bilinear and $n = 3$ for trilinear interpolation. Here, we

will assume that we have known data at the 2^n endpoints of a n-dimensional regular grid $\cup_{i=1}^{n}[x_{i,1}, x_{i,2}]$. In the standard Euclidean case, multi-dimensional interpolation is defined at a point $\boldsymbol{x} = [x_1, \ldots, x_n]$ as

$$\boldsymbol{\psi}(\boldsymbol{x}) = \sum_{j=1}^{2^n} w_j(\boldsymbol{x})\boldsymbol{\psi}(\boldsymbol{x}_j) \Rightarrow \sum_{j=1}^{2^n} w_j(\boldsymbol{x})\overrightarrow{\boldsymbol{\psi}(\boldsymbol{x})\boldsymbol{\psi}(\boldsymbol{x}_j)} = \mathbf{0},$$

where $\boldsymbol{x}_j = [(x_j)_1, \ldots, (x_j)_n]$ are the coordinates of the endpoints, $\boldsymbol{\psi}(\boldsymbol{x}_j)$ are the corresponding values and $w_j(\boldsymbol{x}) = \prod_{i=1}^{n}\left(1 - \frac{|x_i - (x_j)_i|}{x_{i,2} - x_{i,1}}\right)$. For a Riemannian manifold $\boldsymbol{\Psi}$, the equivalent expression is

$$\sum_{j=1}^{2^n} w_j(\boldsymbol{x}) \log_{\boldsymbol{\psi}(\boldsymbol{x})} \boldsymbol{\psi}(\boldsymbol{x}_j) = \mathbf{0}.$$

Notice its similarity to the weighted mean in Eq. (7). Therefore, the solution is defined as

$$\arg\min_{\boldsymbol{\psi}(\boldsymbol{x})\in\boldsymbol{\Psi}} \sum_{j=1}^{2^n} w_j(\boldsymbol{x})\text{dist}(\boldsymbol{\psi}(\boldsymbol{x}), \boldsymbol{\psi}(\boldsymbol{x}_j))^2.$$

Since interpolation is an operation on a fixed grid, the optimization is over $\boldsymbol{\psi}$ and not over the fixed $\boldsymbol{x}$'s. We use Algorithm 1 to compute the interpolated values.

Interpolation on a non-regular grid

Finally, we consider the case where we have m known measurements $\{\boldsymbol{\psi}(\boldsymbol{x}_j)\}_{j=1,\ldots m}$ scattered on a non-regular grid $\{\boldsymbol{x}_j\}_{j=1,\ldots,m}$. We can either use simple nearest-neighbor interpolation or we can estimate the unknown values via the inverse distance weighting (IDW) method [28]. That is, we define the interpolation as the solution to $\arg\min_{\boldsymbol{\psi}(\boldsymbol{x})\in\boldsymbol{\Psi}} \sum_{j=1}^{m} w_j \text{dist}(\boldsymbol{\psi}(\boldsymbol{x}), \boldsymbol{\psi}(\boldsymbol{x}_j))^2$, where $w_j(\boldsymbol{x}) = \frac{K(\boldsymbol{x},\boldsymbol{x}_j)^{-p}}{\sum_{i=1}^{m} K(\boldsymbol{x},\boldsymbol{x}_j)^{-p}}$ is a simple IDW weighting function as defined in [28], K is a given metric operator (distance) from $\boldsymbol{x}_j$ to $\boldsymbol{x}$, and p is a positive real number, called the power parameter.

3.2. Convolution and Filtering of ODFs

Convolution is the cornerstone of many data processing applications such as denoising, smoothing, edge detection, image sharpening and filtering. One of the best known applications will be that of Gaussian smoothing that is commonly used to remove noise from data. While this is a simple and robust filtering technique commonly used in various applications, it suffers from the problem of blurring and mislocating discontinuities in the data. These discontinuities usually correspond to important structures of the data, such as edges and corners in images. As the isotropic Gaussian kernel is the solution to the linear diffusion equation, Gaussian convolution is equivalent to linear diffusion. In particular, Gaussian smoothing corresponds to having the same diffusivity constant c independently of the spatial coordinates $\boldsymbol{x}$. Therefore, various nonlinear diffusion filtering techniques, which smooth the data while preserving discontinuities, have been proposed. The main idea behind these anisotropic filtering methods is to make use of a non-homogeneous diffusivity functions $c(\cdot)$ that depend on the data at $\boldsymbol{x}$. For example, in [26], a diffusivity function $c(\cdot)$ based on the derivative of the data at $\boldsymbol{x}$ is used to control the smoothing near the edges of the image. Thus, diffusion across edges is greatly reduced while being allowed along edges. This prevents edges and corners from being smoothed during the filtering process.

In this section, we will first consider the problem of doing convolution and (isotropic) filtering on a field of ODFs. In the latter part of this section, we will detail how to perform anisotropic filtering. The motivation for filtering diffusion data is that the signal-to-noise ratio in HARDI data is generally low, especially with the high gradient strengths necessary for imaging and resolving crossing white matter tracts and other regions of complex tissue architecture.

Convolution

Traditionally, filtering is based on the idea of convoluting data with different filters $\mathbf{g}$. Discrete convolution of $\boldsymbol{\psi}$ with the function $\mathbf{g}$ is defined as $\boldsymbol{\xi}(\boldsymbol{x}) = \boldsymbol{\psi}(\boldsymbol{x}) * \mathbf{g}(\boldsymbol{x}) = \sum_{\mathbf{u}} \mathbf{g}(\mathbf{u})\boldsymbol{\psi}(\boldsymbol{x}+\mathbf{u})$. Assuming that $\sum_{\mathbf{u}} \mathbf{g}(\mathbf{u}) = 1$ and rewriting, we get $\sum_{\mathbf{u}} \mathbf{g}(\mathbf{u})\overrightarrow{\boldsymbol{\xi}(\boldsymbol{x})\boldsymbol{\psi}(\boldsymbol{x}+\mathbf{u})} = \mathbf{0}$.

When the data lies on $\boldsymbol{\Psi}$, it is immediate to see that the equivalent expression is $\sum_{\mathbf{u}} \mathbf{g}(\mathbf{u}) \log_{\boldsymbol{\xi}(\boldsymbol{x})} \boldsymbol{\psi}(\boldsymbol{x}+\mathbf{u}) = \mathbf{0}$. Again, the solution is defined as

$$\arg\min_{\boldsymbol{\xi}(\boldsymbol{x})\in\boldsymbol{\Psi}} \sum_{\mathbf{u}} \mathbf{g}(\mathbf{u})\text{dist}(\boldsymbol{\xi}(\boldsymbol{x}), \boldsymbol{\psi}(\boldsymbol{x}+\mathbf{u}))^2. \quad (8)$$

When $\mathbf{g}(\mathbf{u}) \geq 0$, the optimization problem in Eq. (8) remains convex, and the convergence to the correct solution is again guaranteed and solved using Algorithm 1. When $\mathbf{g}(\mathbf{u}) < 0$, convergence is guaranteed when the Hessian of the function in Eq. (8) is positive definite.

Anisotropic Filtering

We will first review the well-known problem of minimizing the spatial irregularity of a scalar field $\boldsymbol{\psi} : \mathbb{R}^d \to \mathbb{R}$. Consider the cost function

$$E(\boldsymbol{\psi}) = \int \|\nabla\boldsymbol{\psi}(\boldsymbol{x})\|^2 d\boldsymbol{x}, = \int \sum_{i=1}^{d} \langle \delta_i\boldsymbol{\psi}, \delta_i\boldsymbol{\psi}\rangle d\boldsymbol{x}, \quad (9)$$

where $\nabla\boldsymbol{\psi}(\boldsymbol{x}) = [\delta_1\boldsymbol{\psi}, \ldots, \delta_d\boldsymbol{\psi}]^\top$ is the spatial gradient of $\boldsymbol{\psi}$ and $\nabla = [\delta_1, \ldots, \delta_d]^\top$ the gradient operator. We can compute the gradient of the energy functional by considering a tangent vector $\boldsymbol{\phi}(\boldsymbol{x})$ at $\boldsymbol{\psi}(\boldsymbol{x})$ and evaluating the functional derivative. We have

$$\delta_{\boldsymbol{\phi}} E(\boldsymbol{\psi}) = 2\int \langle \nabla\boldsymbol{\psi}(\boldsymbol{x}), \nabla\boldsymbol{\phi}(\boldsymbol{x})\rangle d\boldsymbol{x}. \quad (10)$$

Now, we want to find the gradient ∇E such that it is independent of the tangent vector $\boldsymbol{\phi}(\boldsymbol{x})$. That is to say, we are trying to find ∇E such that $\delta_{\boldsymbol{\phi}} E(\boldsymbol{\psi}) = \int \langle \nabla E, \boldsymbol{\phi}(\boldsymbol{x})\rangle d\boldsymbol{x}$. This is done by assuming homogeneous Neumann boundary

conditions and applying integration by parts. We get

$$\delta_{\boldsymbol{\phi}} E(\boldsymbol{\psi}) = -2 \int \langle \Delta \boldsymbol{\psi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x}) \rangle d\boldsymbol{x}, \tag{11}$$

where $\Delta \boldsymbol{\psi}(\boldsymbol{x}) = \sum_{j=1}^{d} \delta_j^2 \boldsymbol{\psi}$ is the Laplacian operator. Therefore, the gradient of the energy is $\nabla E = -2\Delta\boldsymbol{\psi}(\boldsymbol{x})$. Using gradient descent, this brings us to the familiar evolution equation

$$\boldsymbol{\psi}_{t+1}(\boldsymbol{x}) = \boldsymbol{\psi}_t(\boldsymbol{x}) - \gamma \nabla E = \boldsymbol{\psi}_t(\boldsymbol{x}) + 2\gamma \Delta \boldsymbol{\psi}(\boldsymbol{x}), \tag{12}$$

where $\gamma > 0$.

For anisotropic filtering of a scalar field, the cost function to be minimized is such that it penalizes smoothing across edges, i.e., when the derivative of the scalar field is large. Usually, this is done via a penalizing function and an example of such a function is $c(x) = \exp(-\frac{x^2}{\kappa^2})$. In that case, the discrete implementation for anisotropic filtering is a straightforward variation of Eq. (12), where one penalizes contribution of $\Delta_{\mathbf{u}} \boldsymbol{\psi}(\boldsymbol{x})$ in the spatial direction $\mathbf{u}$ according to the norm of the gradient $\|\delta_{\mathbf{u}} \boldsymbol{\psi}(\boldsymbol{x})\|$ in the direction $\mathbf{u}$. That is,

$$\nabla E = -2c(\|\delta_{\mathbf{u}} \boldsymbol{\psi}(\boldsymbol{x})\|) \Delta_{\mathbf{u}} \boldsymbol{\psi}(\boldsymbol{x}). \tag{13}$$

Now, for a field of ODFs, and making use of the intrinsic gradient descent scheme, the evolution equation becomes

$$\begin{aligned} \boldsymbol{\psi}_{t+1}(\boldsymbol{x}) &= \exp_{\boldsymbol{\psi}_t(\boldsymbol{x})}(-\gamma \nabla E) \\ &= \exp_{\boldsymbol{\psi}_t(\boldsymbol{x})}(2\gamma \Delta \boldsymbol{\psi}(\boldsymbol{x})), \end{aligned} \tag{14}$$

where $\gamma > 0$, where $\Delta\boldsymbol{\psi}(\boldsymbol{x})$ is the Laplacian operator on the space $\boldsymbol{\Psi}$. Similar to what was done for DTI [19], we define the equivalent of Eq. (9) as

$$E(\boldsymbol{\psi}) = \int \sum_{i,j}^{d} \langle \delta_i \boldsymbol{\psi}(\boldsymbol{x}), \delta_j \boldsymbol{\psi}(\boldsymbol{x}) \rangle_{\boldsymbol{\psi}(\boldsymbol{x})} d\boldsymbol{x},$$

where $\delta_i \boldsymbol{\psi}(\boldsymbol{x}) \in T_{\boldsymbol{\psi}(\boldsymbol{x})} \boldsymbol{\Psi}$ is the partial derivative of $\boldsymbol{\psi}$ in direction i. The gradient of the energy is then

$$\nabla E = -2 \sum_{i,j} \Delta_{ij} \boldsymbol{\psi}(\boldsymbol{x}), \qquad \text{where } \Delta_{ij} = \delta_i \delta_j.$$

Therefore, to perform the anisotropic filtering of field of ODFs, we proceed as in Eq. (13), and modify Eq. (14) to account for the penalizing function $c(\cdot)$.

Finally, we will show how to approximate the operator Δ_{ij} for a given field of ODFs $\boldsymbol{\psi}$. To this end, recall that under the finite central difference methods for functions, we have the following expressions for the Euclidean derivatives, with $\{\mathbf{e}_i\}$ being the orthonormal vectors of the canonical basis:

$$\delta_i \boldsymbol{\psi}(\boldsymbol{x}) = \frac{1}{\epsilon}(\boldsymbol{\psi}(\boldsymbol{x} + \frac{\epsilon \mathbf{e}_i}{2}) - \boldsymbol{\psi}(\boldsymbol{x}) + \boldsymbol{\psi}(\boldsymbol{x}) - \boldsymbol{\psi}(\boldsymbol{x} - \frac{\epsilon \mathbf{e}_i}{2})),$$

$$\delta_i^2 \boldsymbol{\psi}(\boldsymbol{x}) = \frac{1}{\epsilon^2}(\boldsymbol{\psi}(\boldsymbol{x} + \epsilon \mathbf{e}_i) - 2\boldsymbol{\psi}(\boldsymbol{x}) + \boldsymbol{\psi}(\boldsymbol{x} - \epsilon \mathbf{e}_i)),$$

$$\delta_i \delta_j \boldsymbol{\psi}(\boldsymbol{x}) = \frac{1}{\epsilon^2}(\boldsymbol{\psi}(\boldsymbol{x} + \frac{\epsilon \mathbf{e}_i}{\sqrt{2}} + \frac{\epsilon \mathbf{e}_j}{\sqrt{2}}) - \boldsymbol{\psi}(\boldsymbol{x} + \frac{\epsilon \mathbf{e}_i}{\sqrt{2}} - \frac{\epsilon \mathbf{e}_j}{\sqrt{2}}) - \boldsymbol{\psi}(\boldsymbol{x} - \frac{\epsilon \mathbf{e}_i}{\sqrt{2}} + \frac{\epsilon \mathbf{e}_j}{\sqrt{2}}) + \boldsymbol{\psi}(\boldsymbol{x} - \frac{\epsilon \mathbf{e}_i}{\sqrt{2}} - \frac{\epsilon \mathbf{e}_j}{\sqrt{2}})).$$

Therefore, for the first and second order derivatives in Riemannian space $\boldsymbol{\Psi}$, we have the following approximations:

$$\delta_i \boldsymbol{\psi}(\boldsymbol{x}) \approx \frac{1}{\epsilon}(\overrightarrow{\boldsymbol{\psi}(\boldsymbol{x})\boldsymbol{\psi}(\boldsymbol{x} + \frac{\epsilon \mathbf{e}_i}{2})} - \overrightarrow{\boldsymbol{\psi}(\boldsymbol{x})\boldsymbol{\psi}(\boldsymbol{x} - \frac{\epsilon \mathbf{e}_i}{2})}),$$

$$\delta_i^2 \boldsymbol{\psi}(\boldsymbol{x}) \approx \frac{1}{\epsilon^2}(\overrightarrow{\boldsymbol{\psi}(\boldsymbol{x})\boldsymbol{\psi}(\boldsymbol{x} + \epsilon \mathbf{e}_i)} + \overrightarrow{\boldsymbol{\psi}(\boldsymbol{x})\boldsymbol{\psi}(\boldsymbol{x} - \epsilon \mathbf{e}_i)},$$

$$\delta_i \delta_j \boldsymbol{\psi}(\boldsymbol{x}) \approx \frac{1}{\epsilon^2}(\overrightarrow{\boldsymbol{\psi}(\boldsymbol{x})\boldsymbol{\psi}(\boldsymbol{x} + \frac{\epsilon \mathbf{e}_i}{\sqrt{2}} + \frac{\epsilon \mathbf{e}_j}{\sqrt{2}})} - \overrightarrow{\boldsymbol{\psi}(\boldsymbol{x})\boldsymbol{\psi}(\boldsymbol{x} + \frac{\epsilon \mathbf{e}_i}{\sqrt{2}} - \frac{\epsilon \mathbf{e}_j}{\sqrt{2}})} + \overrightarrow{\boldsymbol{\psi}(\boldsymbol{x})\boldsymbol{\psi}(\boldsymbol{x} - \frac{\epsilon \mathbf{e}_i}{\sqrt{2}} - \frac{\epsilon \mathbf{e}_j}{\sqrt{2}})} - \overrightarrow{\boldsymbol{\psi}(\boldsymbol{x})\boldsymbol{\psi}(\boldsymbol{x} - \frac{\epsilon \mathbf{e}_i}{\sqrt{2}} + \frac{\epsilon \mathbf{e}_j}{\sqrt{2}})}).$$

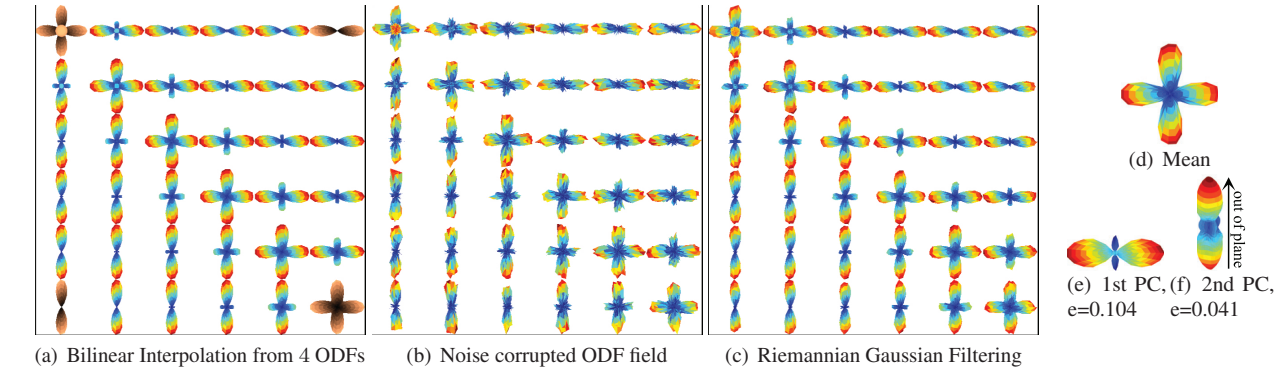
The derivation of the various approximations of the derivatives $\boldsymbol{\psi}$ proceeds as in the DTI case [19, 25].

4. Experiments

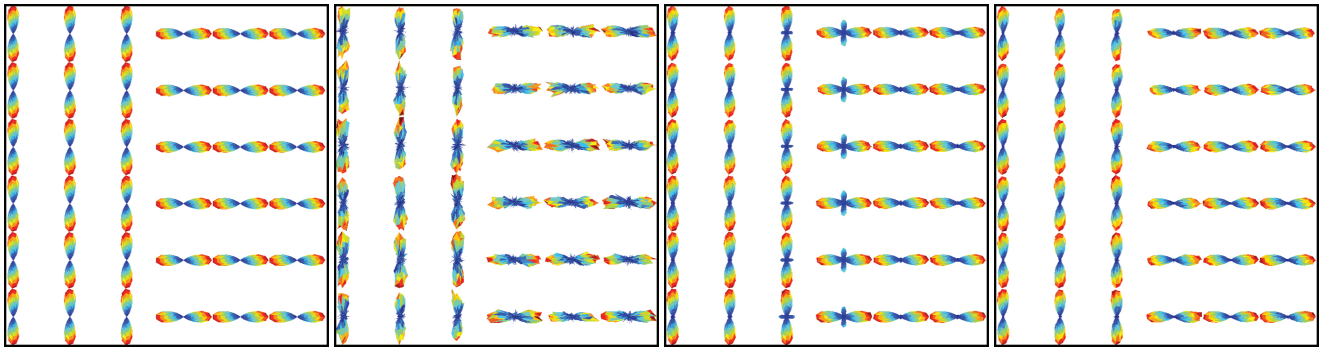
In this section, we present experiments on synthetic and real datasets using the proposed framework. We first illustrate the various operations previously introduced, such as interpolation, filtering, and computation of the mean and principal components (PC) of the synthetic ODF fields in Fig. 1. Fig. 1(a) shows the bilinear interpolation of the four endpoints where the original ODFs are shaded in orange. The top right and bottom left voxels contain ODFs of 1 fiber, the bottom right ODF of 2 fibers and the top left ODF of 3 fibers, with 1 fiber pointing out of the plane. Next, we illustrate Gaussian filtering. We add Gaussian noise (in the Riemannian sense) to the resulting ODF field in Fig. 1(a), as shown in Fig. 1(b). Fig. 1(c) shows the results of Gaussian filtering. Notice that Gaussian filtering managed to remove a significant amount of noise. Figs. 1(d)-1(f) show the Riemannian mean and the first two principal components with eigenvalues $0.104, 0.041$ of the ODF field from Fig. 1(a). The first PC is oriented from left to right, whereas the second PC is oriented going out of the plane.

Next, we perform anisotropic filtering (AF) on a ODF field with a sharp discontinuity, as shown in Fig. 1(g). The function $c(x) = \exp(-\frac{x^2}{\kappa^2})$ is used in AF, and 30 iterations are performed. Again, we add noise to this ODF field, as shown in Fig. 1(h). Fig. 1(i) shows the results of Euclidean Anisotropic Filtering (EAF), where instead of taking into account the Riemannian structure of the ODFs, we simply treat each ODF as a vector in Euclidean space, and perform anisotropic filtering in the classical Euclidean way. Fig. 1(j) shows the results of Riemannian Anisotropic Filtering (RAF). From Figs. 1(i)-1(j), one can see that RAF is able to respect the discontinuity whereas EAF blurs the edges.

Next, we examine the differences in performance between RAF and EAF as we vary the amount of noise added to the ground-truth ODF field over 100 trials for



(a) Bilinear Interpolation from 4 ODFs (b) Noise corrupted ODF field (c) Riemannian Gaussian Filtering



(g) Synthetic ODFs used for AF (h) Noise corrupted ODF field for AF (i) Euclidean Anis. Filtering (j) Riemannian Anis. Filtering

Figure 1. Illustration of interpolation, Gaussian filtering, statistics computation, and anisotropic filtering of synthetic ODF fields.

Noise level, σ=10%	5σ	4σ	3σ	2σ	σ
Euclidean measure	0.74	0.44	0.30	0.24	0.20
Riemannian measure	0.92	0.78	0.71	0.66	0.63

Table 1. Mean of (Amount of error after RAF)/(Amount of error after EAF) in 100 trials under two different measures.

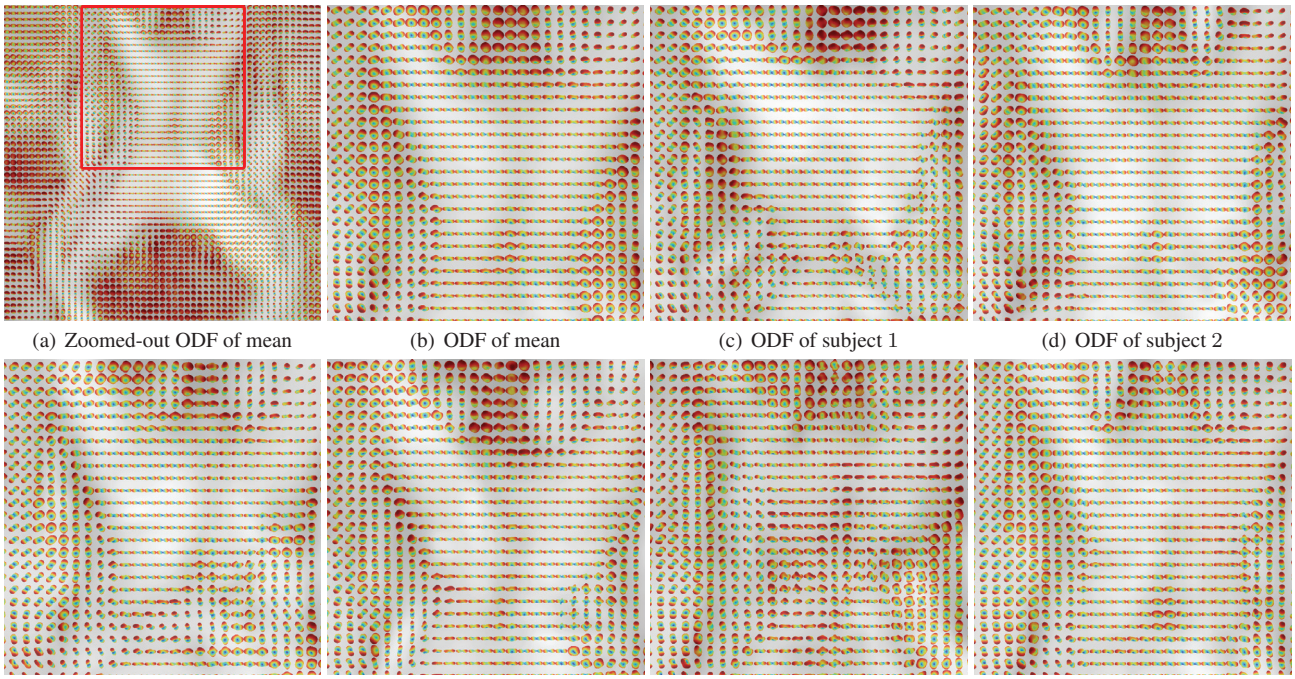
5 different noise levels $\sigma - 5\sigma$, where $\sigma = 10\%$. For both types of AFs, we measure the amount of error from the ground-truth ODF field using two measures, Riemannian $\sum_{\boldsymbol{x}} \| \log_{\boldsymbol{\psi}_{truth}(\boldsymbol{x})}(\boldsymbol{\psi}_{filtered}(\boldsymbol{x}))\|_{\boldsymbol{x}}$ and Euclidean $\sum_{\boldsymbol{x}} \|\boldsymbol{\psi}_{truth}(\boldsymbol{x}) - \boldsymbol{\psi}_{filtered}(\boldsymbol{x})\|$ dissimilarity. In each trial, we iterate AF 30 times. Table 1 shows the ratio of error for RAF to EAF under the Euclidean and Riemannian measures. RAF always results in a lesser amount of error compared to EAF. In addition, as the amount of noise that is added to the ODF field increases and the discontinuity becomes blurred, we see that the error in RAF converges to that of EAF.

Finally, we apply our Riemannian framework to a subset of a HARDI human brain database [8] of 12 subjects. Diffusion weighted MR images were obtained using the following imaging parameters: 21 axial slices (5 mm thick), TR/TE=6090/91.7 ms, with a 128×128 acquisition matrix (1.8 mm in-plane resolution). 3 with no diffusion sensitization and $n = 27$ diffusion weighted images at $b = 1132$ s/mm^2 were acquired. Gradient directions were evenly distributed on the hemisphere. The ODFs of the 12 subjects were fluidly registered to a target template and transformed to isotropic voxel resolution ($128 \times 128 \times 93$ voxels with $1.7 \times 1.7 \times 1.7$ mm^3 resolution) [8]. We will first use our framework to calculate the mean ODF field of the 12 subjects. A portion of the results in an axial slice near the corpus callosum are shown in Fig. 2. Next, we apply our Riemannian anisotropic filtering method to the mean ODF and the ODF fields of the 12 subjects. Fig. 3 shows a portion of the results of RAF and Fig. 3(a)-3(b) show the resulting ODF fields after applying RAF to the ODF fields of the calculated mean and subject 1. Again, the function $c(\cdot)$ used in RAF is a Gaussian function, and 30 iterations are done. Notice that RAF is able to smooth the ODF field sufficiently but still preserves the discontinuities between different bundles.

5. Conclusions

We present a Riemannian framework for the processing of ODFs that does not require that the ODF be represented by any parametric model. The various Riemannian operations are in closed form and easily and efficiently computed. Results on synthetic and real data demonstrate and quantify the advantage of working with our proposed framework as opposed, for instance, to a Euclidean approach. Choosing the correct metric for ODFs remains an open question and we present a rigorous and well founded framework with desirable properties that the other metrics do not have.

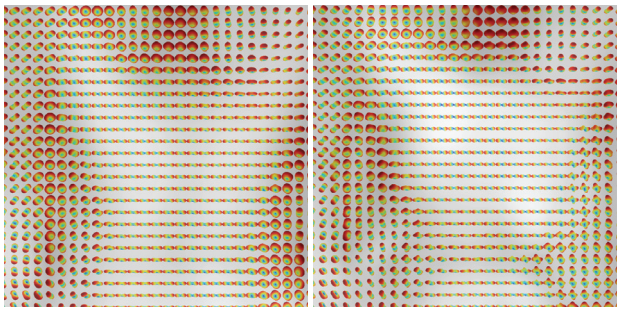
Acknowledgments. We would like to thank Ming-Chang Chiang for providing us with the registered datasets. Work supported by startup funds from JHU, by grants NSF CAREER IIS-0447739, NIH R01 HD050735, NIH R01 EB007813, NIH P41 RR008079, NIH P30 NS057091, ONR N00014-05-10836 and ONR N00014-09-1-0084.



(a) Zoomed-out ODF of mean (b) ODF of mean (c) ODF of subject 1 (d) ODF of subject 2

(e) ODF of subject 3 (f) ODF of subject 4 (g) ODF of subject 5 (h) ODF of subject 6

Figure 2. Calculation of mean ODF field from 12 subjects. Fig. 2(a) show the mean ODF field of one slice where the zoomed-in results of the red box shown in Fig. 2(b). Figs. 2(c)-2(h) show ODF fields of a sample of 6 different subjects.



(a) ODF of mean after RAF (b) ODF of subject 1 after RAF

Figure 3. Anisotropic filtering on brain data. Figs. 3(a)-3(b) show the ODF fields in Figs. 2(b) and 2(c) after applying RAF.

References

[1] S. Amari. *Differential-Geometrical Methods in Statistics*. Springer, 1985.

[2] V. Arsigny et al. Log-Euclidean metrics for fast and simple calculus on diffusion tensors. *MRM*, 56:411–421, 2006.

[3] A. Barmpoutis et al. Symmetric positive 4th order tensors and their estimation from diffusion weighted MRI. In IPMI, pages 308–319, 2007.

[4] P. Basser et al. Estimation of the effective self-diffusion tensor from the NMR spin echo. *Journal of Magnetic Resonance B*, 103:247–254, 1994.

[5] P. Basser et al. In vivo fiber tractography using DT-MRI data. *MRM*, 2000.

[6] M. Berger. *A Panoramic View of Riemannian Geometry*. Springer, 2003.

[7] N. N. Cencov. Statistical decision rules and optimal inference. In *Translations of Mathematical Monographs*, volume 53. AMS, 1982.

[8] M.-C. Chiang et al. Brain fiber architecture, genetics, and intelligence: A high angular resolution diffusion imaging (HARDI) study. In *MICCAI*, 2008.

[9] M. Descoteaux et al. Regularized, fast and robust analytical q-ball imaging. *MRM*, 58:497–510, 2007.

[10] M. Descoteaux et al. High angular resolution diffusion MRI segmentation using region-based statistical surface evolution. *JMIV*, 2008.

[11] P. T. Fletcher and S. Joshi. Riemannian geometry for the statistical analysis of diffusion tensor data. *Signal Processing*, 87(2), 2007.

[12] L. R. Frank. Characterization of anisotropy in high angular resolution diffusion-weighted MRI. *MRM*, 47(6):1083–1099, 2002.

[13] M. Frechet. Les elements aleatoires de nature quelconque dans un espace distancie. *Annales De L'Institut Henri Poincare*, 10:235–310, 1948.

[14] A. Ghosh et al. Riemannian framework for estimating symmetric positive definite 4th order diffusion tensors. In *MICCAI*, 2008.

[15] A. Goh and R. Vidal. Unsupervised Riemannian clustering of probability density functions. In *ECML PKKD*, 2008.

[16] P. Hagmann et al. DTI mapping of human brain connectivity: statistical fibre tracking and virtual dissection. *Neuroimage*, 19(3):545–554, 2003.

[17] H. Karcher. Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics*, 30(5):509–541, 1977.

[18] G. Kindlmann et al. Geodesic-loxodromes for diffusion tensor interpolation and difference measurement. In *MICCAI*, 2007.

[19] K. Krajsek et al. Riemannian anisotropic diffusion for tensor valued images. In *ECCV*, 2008.

[20] S.-M. Lee et al. Dimensionality reduction and clustering on statistical manifolds. *CVPR*, 2007.

[21] C. Lenglet et al. Statistics on the manifold of multivariate normal distributions: Theory and application to diffusion tensor MRI processing. *JMIV*, 2006.

[22] T. McGraw et al. Segmentation of high angular resolution diffusion MRI modeled as a field of von Mises-Fisher mixtures. In *ECCV*, 2006.

[23] W. Mio, D. Badlyans, and X. Liu. A computational approach to Fisher information geometry with applications to image analysis. In *EMMCVPR*, 2005.

[24] E. Özarslan et al. Generalized DTI and analytical relationships between DTI and high angular resolution diffusion imaging. *MRM*, 50:955–965, 2003.

[25] X. Pennec et al. A Riemannian framework for tensor computing. *IJCV*, 66(1):41–46, 2006.

[26] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *PAMI*, 12(7):629–639, 1990.

[27] C. R. Rao. Information and accuracy attainable in the estimation of statistical parameters. *Bull. Calcutta Math Soc.*, 37:81–89, 1945.

[28] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proc. of ACM national conference*, pages 517–524. ACM, 1968.

[29] A. Srivastava et al. Riemannian analysis of probability density functions with applications in vision. In *CVPR*, 2007.

[30] D. S. Tuch. High angular resolution diffusion imaging reveals intravoxel white matter fiber heterogeneity. *MRM*, 48:577–582, 2002.

[31] D. S. Tuch. Q-ball imaging. *MRM*, 52(6):1358–1372, 2004.